Universidad Carlos III de Madrid

Proyecto Fin de Carrera
Grado En Ingenieria Informatica

Assesing the Effects of Blocking Strategies in the Online Advertising

---

Autor:
Pelayo Vallina Rodriguez

Tutor:
Ruben Cuevas Rumin

June 22, 2016

Acknowledgements

I want to express my gratitude to my family. They have supported me all my life. My brother, Narseo, has been a fountain of inspiration for me.

I would like to thank Dr. Roberto Gonzalez to give me the opportunity to complete my internship at NEC labs and to Miriam Marciel for helping me whenever I needed support.

Finally, I would like to thank all the people working to protect our digital rights.

Pelayo Vallina Rodriguez

Abstract

This project falls in the area of online privacy, in particular in the impact of online tracking on user's privacy and its importance on today's web, greatly fueled by advertising revenues.

During this work, I have implemented a plugin for Firefox that helps users to prevent abuses inflicted by the advertising industry. The plugin enables users to select their own rules using a graphic web interface, where they can create their own rules or select or to define under which circumstances they want to prevent tracking activities in order to stay in control of their data. I complement my development efforts with a research study that characterizes how HTTP Cookies are used to build user profiles and how they affect the ads printed on websites visited by the user.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

It is well known that the World Wide Web is fueled by advertising revenues. This economic model allows Internet users to access free content and to benefit from online services without paying for a subscription fee. However, this model takes a toll on web users in terms of degraded web experience (due to intrusive ads) and also in terms of privacy violations caused by online tracking.

Modern websites use "web cookies" [1] – a piece of data that web services use to store information of the users in the browser – for tracking users on the web. As a result, Internet users have the feeling that a number of unknown online companies know every single detail about them: from their preferred clothing brand to their sexual preference. Unfortunately, we, as users, rarely know the companies tracking us when we visit a given website beyond the big players like Google or Facebook. Furthermore, we do not know how they manage the information they collect about us, whether they share it with third-parties and if there's any hidden purpose behind their tracking activity. In fact, user tracking in the web takes place without user awareness and explicit consent in most cases as widely reported by the media in the last few years. As a result, the society has gradually became more privacy-aware, pushing regulatory bodies in the European Union to legislate the

online advertisement ecosystem in order to protect user rights.

Social demands for online privacy has given birth to adblocking plugins like Ad Block Plus [5]. These plugins, which run on the user browser, are now used by millions of users hoping to protect their privacy or to avoid the presence of intrusive ads embedded in websites. However, according to a study conducted by Adobe and PageFair – a company that provides web publishers with technologies to avoid the effects of ad-blocker [4] – more than 50% of ad-blocker users take advantage of these technologies to protect their online privacy and security [3]. According to this study, 45 million and 77 million of American and European web surfers use adblocking technologies respectively.

The increasing presence of ad-blockers on user's browsers has a direct impact on the revenues of web services and advertising revenues, putting at risk the economic model that has sustained the web as we know it. This creates a tension between users and the multiple organizations and entities that benefit from ads. Online publishers and ad networks – an industry that employed more than 3.4 million people in Europe in 2012 according to the Interactive Advertising Bureau (IAB) – lost $21.8B in advertisement revenues [3]. Both the economic value of the online advertising industry and the number of job positions that it generates are solid arguments to keep this industry alive and to find a solution that achieves a balance between user's privacy and advertising revenues. However, online advertising is here to stay. The industry developed technologies such as PageFair's anti-adblocking javascripts in response to the increasing presence of ad-blockers. Publishers take advantage of these technologies not only to cirvunvent the presence of ad-blockers but also to to educate the user about the impact of running such plugins, or to ask them for a subscription fee to access the content [4].

Nevertheless, not all web ads have a negative impact on user's privacy nor interfere with user's web browsing experience by displaying intrusive ads (e.g., pop-ups). Unfortunately, today's ad-blockers do not make any difference between benign ads or intrusive ones; they follow a radical approach considering all of them as part of the same bucket: they simply block them all. Therefore, it is imperative to find technological solutions and appropriate policies that can guarantee a sustainable web without violating user's right

to privacy and anonimity.

This goal has been the main driver of my work. In this dissertation, I present a solution that seeks to achieve a sustainable web and advertising model that guarantees user's right to anonymity. In that effort, I designed and implemented a browser plugin, which we plan to release publicly for Mozilla Firefox, that achieves this goal. Furthermore, I extend my development efforts with a research study that investigates the behavior of the online advertising industry and the role played by two different tracking activities in order to deliver targetted ads: web cookies and online trackers.

My dissertation is part of the broader european project "Towards transparency and privacy in the online advertising business" (TYPES)[7]. This is a joint effort between institutions like NEC Labs Europe, Telefonica I+D, Open University of Israel, IMDEA Networks and Eurecat among other partners [8]. I completed this work during my internship at NEC Labs Europe in Heidelberg between January and July 2016.

1.2 Project Goals

This project has two different but complementary goals:

1. Development of a browser plugin for Mozilla Firefox: The first aim of this project is developing a browser plugin to protect the user's privacy and to minimize their risks of being tracked. This plugin enables users to block trackers and intrusive ads at the traffic level according with a number of predefined rules. These predefined rules block the principal trackers that exist nowadays in the web sites, like *doubleClick.net*. However, as opposed to state of the art ad-blockers like AdBlock Plus, my plugin also gives users the opportunity to choose the trackers and organizations that they want to block. The reasoning behind this design feature is that some users may not care about being tracked in some websites because they don't feel that their privacy is at risk. In fact, users can benefit from such tracking activity as they can receive ads about products and services of their interest or to obtain economic benefits through price-discrimination . Both the pre-defined rules

and the user-specific rules are centrally stored in a web service [1]. The plugin communicates with the web site using an authorization protocol, Oauth2, to get the user's rules.

2. A study of the role that HTTP cookies plays in Online Behavioral Advertising(OBA). In that part of the project, I analyze empirically the multiple privacy implications that today's online advertising model has on user's privacy. In this study, I explore how the information embedded on web cookies defines the type of advertisement delivered to the user when visiting different websites about different topics. This empirical research study is supported by the data provided by the plugin described in Goal 1. To get this goal I have selected a different training pages that I will explain in chapter 5 *Study OBA* and I will present the results of this study in chapter 6 *Analysis*.

1.3 Dissertation structure

The rest of the dissertation is structured as follows:

- **Background:** In this chapter, I introduce the basic concepts and terms in the area of online privacy. I will emphasize the techniques used to perform user tracking and the privacy concerns that arise in the context of the web browser.
- **State of Art:** In this section, I will discuss previous research efforts in the area of online privacy and the different solutions that exist as of today to protect user's privacy on the web. In this chapter I will also introduce the European legislation and the ongoing regulatory efforts to protect user's online privacy.
- **Implementation:** This section describes the browser plugin that I have developed in this project (Goal 1). I highlight the programming languages, equipment, software design and the use case that I have used during my development efforts.

[1] `http://typesbackend.newlab2.wedia.gr/login`

- **Analysis of the impact of web cookies on OBA:** In this chapter, I describe the tools for the analysis of web cookies (Goal 2) and the methodology that I have followed in my research efforts. In my research study, I create 120 different user profiles (e.g., personas) leveraging the cookies collected with the plugin. With those personas, I advance the state of the art in analyzing the effect of web cookies on Online Behavioral Advertising (OBA). I characterize the effect of web cookies on the delivery of targetted advertisements for 10 relevant websites and how it affects the ads delivered on a entirely neutral website.

- **Planification and Budget:** This chapter concludes the dissertation and shows how I planned the project and the budget required to fully implement it.

- **Conclusions and Future Work:** In this section, I summarize the conclusions of this project and discuss future research directions that will complement the tools and the empirical results presented in this dissertation.

Chapter 2

Background

This chapter presents the basic background concepts required to understand the web advertising industry, how they perform user tracking and the two type of technology available for users to prevent tracking. This chapter will help the reader to better contextualize the contributions of this project.

2.1 Online Advertising

The online advertising industry is a complex ecosystem composed by multiple entities, each of which plays an specific role in the process of displaying an ad on a website:

- Customer: The customer is the "user" of the web site who receives the ads.
- Publisher: The publisher is the owner of the website that offers space where to put or impress ads.
- Ad company: A company that specializes in offering ads from third companies in web sites.
- Supply-Side Platform (SSP): It is a platform that manages the web sites where ad companies can show their ads. The SSP recieves and

aggregates information about the user's fingerprint, which it later sends to the Ad exchange platform in order to create a new bidding (see below the description of the bidding model).

- Ad Exchange: An ad exchange is a platform that collects websites in which to embed ads and the user's fingerprint from an SSP. The ad exchange is responsible to create biddings for printing ads in these these sites and sends these biddings to the DSP.

- Demand-Side Platform (DSP): DSP is a technology platform for advertisers and ad companies that receives information from the potential user of the web site and the bidding that ad exchange has created. In this platform advertisers and ad companies bid to publish their ads according to the user's profile.

As we can see, user's information spreads through multiple companies without their awareness. The multi-agent nature of mobile advertisement puts in context how vulnerable user's data is whenever the user visits a website. However, this dissemination depends greatly on the model used to impress an ad on a website or publisher:

- Direct sale: This model, illustrated in Figure 2.1, was common in the early days of the online advertising industry. In that case, the company that wants to advertise its products inserts directly its ads on the space offered by the publisher.

- Trading in ad networks: In this model, ilustrated in Figure 2.1, a company that wants to advertise its products acquires the services of an ad company (which acts as an intermediate between the publisher and the company advertising their products) that creates and distributes their ads by a network of publishers.

- Real-Time Bidding(RTB): This is the most common online advertising model as of today, but also the most complex. As we can see in Figure 2.1, when a user visits a web site that offers ads using this technique, the publisher sends immediately a request to the SSP to offer a slot in which advertisers can print ads. This platform combines the

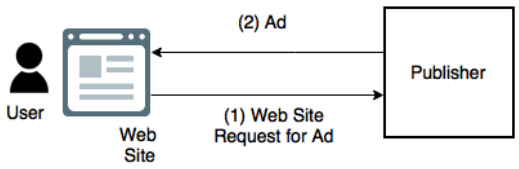

Figure 2.1: Direct Sale.

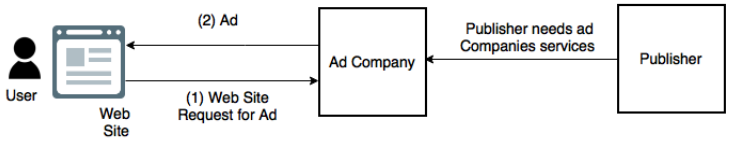

Figure 2.2: Trading in ad networks.

information from the user generated by tracking techniques – in general using web cookies – and the information about the web site where advertisers can embed ads. Then, the SSP sends the offer to the ad exchange which creates a bid that is later announced in the DSP so that advertisers and ad companies can bid up. The winning advertiser publishes the ad in the web site.

2.2 Targeted advertising

Online Behavioural targeted Adversting (OBA) is a technique that enables online advertisers (i.e., companies dedicated to offer advertise on websites), to deliver more relevant ads to interested audiences. This technique takes advantage from user's web fingerprint – left by the user when surfing the web, typically via "web cookies" [1] – to infer user's preferences and taste with the hope to increase user interest on the product.

A web cookie (which I will refer as *"cookie"* for simplicity for the rest of the dissertation) is a small piece of data that web services use to store state of the users in the browser. Given their semi-persistent nature – they are

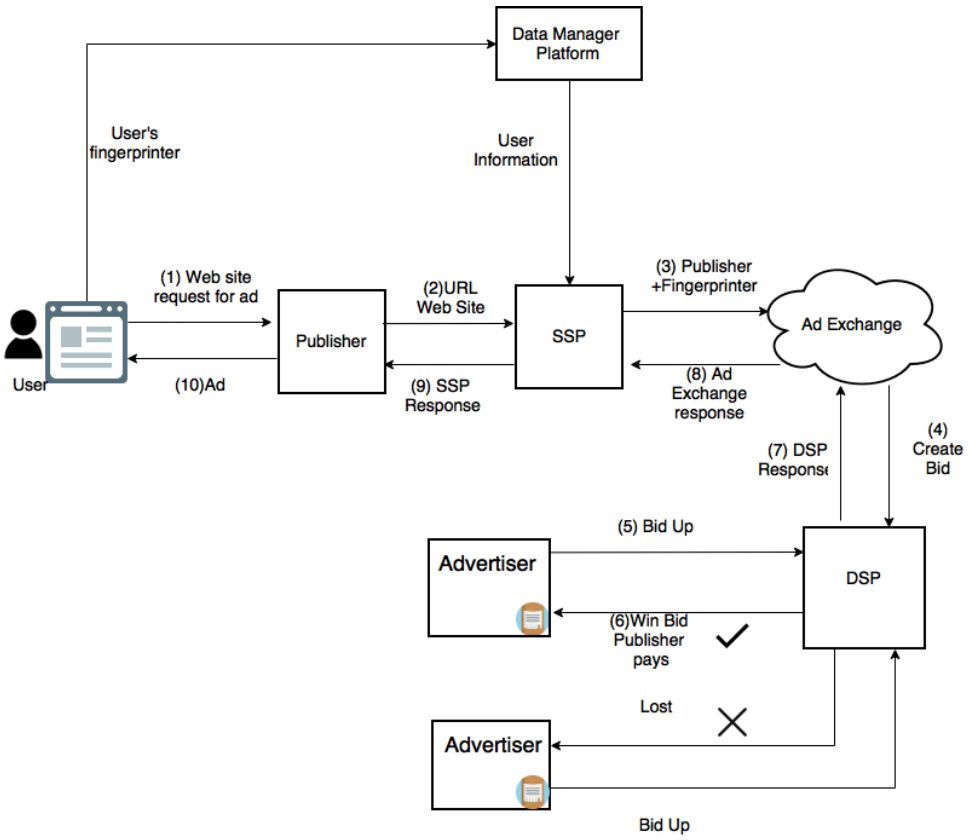

Figure 2.3: Real-Time Bidding(RTB) model.

only deleted if the user explicitely deletes the web cookies via their browser settings – they can reveal a lot of ifnormation about user's activity on the web. They can serve multiple purposes which we will describe in the following chapters. A typical web cookie has the following format:

{Name=*Value*; [expires=*Expiration*;][path=*Path*;][domain=*Domain*;][secure]}

For instance, the cookie below is a real cookie set by `doubleclick.com` that gathers information about the number of times that a user has already seen a given ad. It therefore aims to increase the diversity of ads seen by a given user on a given domain or publisher.

__gads=ID=1c8cc34442991d1d:T=1459865866:S=ALNI_MZKpj2NgtpOkvL-eN2f43wHKWOJQw

I classify web cookies in different categories according to the following criterion:

- By the organization setting the cookie:
  - **First-party cookie:** When a cookie is set by the domain of web server that the user is visiting. For example, if a user is visiting `http://www.bbc.com`, I consider any cookie set by `bbc.com` or any of its sub-domains (e.g.,`iplayer.bbc.com`) as a first-party cookie.
  - **Third-party cookie:** When a cookie is set by a web element or domain wich is not directly associated with the domain providing the content of interest to the user as ad-networks, analytics services or any other tracking included by the web developer on its website. For example, if a user is visiting `http://www.bbc.com`, I consider any cookies set by domains such as `googlesyndication.com` or `doubleclick.com` – two of the most common domains associated with online advertising, both owned by Google – as third-party cookies.
- By its lifespan:

- **Session cookie:** The goal of this cookie is to recognize users uniquely in a web site and rememeber them from page to page. However, they last for as long as the browser session does. Once the session is finished, the cookie is deleted by the browser.
- **Persistent cookie:** The goal of this cookie is to recognize the users and keep that information for future sessions. This web cookie remains stored in the browser until the user actively deletes it or after a timeout is triggered.

2.3 Adblockers and anti-Trackers

Nowadays, web users can use a number of tools to protect their private data and to block any tracking activity. However, not all of technologies work equally neither pursue the same goals. We can divide these tools in two groups according to their goals:

1. Adblockers: The purpose of adblockers is blocking ads in websites. Traditionally, an ad-blocker implements binary filters: either they block all the advertising activity or not. They do not give users the opportunity to configure in which websites they do not want to be tracked or the type of ad they want to see.

2. Anti-trackers: Anti-trackers block any tracking activity beyond ad-related traffic by blocking active scripts that can potentially track user's activity such as JavaScript, Flash plugins or Java applets. They typically allow users to choose which trackers or websites they want to block or in which domain they do not want to be tracked.

In the next chapter, I will provide examples for each type of tool and I will detail how they operate.

Chapter 3

State of the Art

3.1 European Legal Framework

In this section I explain the data protection laws that exist nowadays.

3.1.1 Handbook on European data protection law

In 1948, the United Nations established the right to protection of personal data in the Article 12 of the Universal Declaration of Human Rights (UDHR) [10]. According to the European data protection law, "we consider as private data all the information that refers to an individual person that can identify directly an individual or if this data describes an person in a way which makes possible to find out who is the individual" [11].

We can classify this data in two different groups, sensitive-data and no sensitive-data. Sensitive-data is is any data that can potentially reveal racial or ethnic origin, political opinions, religious beliefs and data concerning health or sexual preferences. According to the law, a tracker should not be present in any web site that covers any of these sensitive topics.

Furthermore, the law says that any user-related data stored in online servers

has to be “anonymised”. It means that it can not be private data – according to the definition that I have presented in this section – . However, when a publisher offers ads, they may use and expose user data that is not always anonymised. According to the study by [12], ad companies use the IP Adress to offer ads. So, if they know our IP addres and our behavioural trait , they can identify us.

Also, there are initiatives fro the ad industry to improve the quality of ads that appear in the web sites. In “Acceptable ads manifesto” [13], ad companies show their compromise to create good ads. However, in this manifesto, they do not talk about track users. For this reason, justice has to take control and prevents these bad practices toughening the law.

3.2 Related Work

As I have exposed, there are laws in the European Union that protect the user private data from the bad practices. However, not all companies obey the law. The study that I expose in this documentation is based in a previos study conducted by Dr. Juan Miguel Carrascosa et al. [23]. The authors of this study present a novel methodology for measuring and understanding OBA [2.2], showing that OBA is a common practice and its presence depends in the value of the user's behavioral trait in the market. Also, they show that even when users visit a sensitive site – web sites related to sexual orientation, health, politics or ethnic origin topics – companies track them, which is illegal.

The study conducted by Barford et al. [24] presents a study about the characteristics and dynamics of online display adverstising and understand the targeting mechanisms that are used by ad serving entities. They developed methods and tools to extract ads from web sites – in this study they use a set of 180 different sites – using 340 different user profiles, that have a specific behavioural trait. The analysis of more than 175,000 ads, shows that 80% of them are the result of targeting mechanism. Only a small fraction of web

sites repeatedly impress the same ad independently of the user's profile.

3.3 Tools to Protect Privacy Data

Nowadays, there are different tools to protect user's private data.

3.3.1 Ad Block Plus

When we talk about blockers, the first name that appears in our minds is AdBlock Plus [5]. Ad Block Plus is a free plugin that supports popular browsers like Chrome, Firefox or Safari.

The goal of this tool is block all intrusive advertisements[1] that appear in the web site, like banners, pop-ups or pop-unders.

Ad Block Plus blocks all these elements using a list of domains that it has to block. Usually a website is made up by different sources, like third parties that offer services to this site or ads from ad companies. When a user visits a web site, Ad Block Plus detects the different sources that appear in the web site. So, if one of these sources match with one of the domains of the list, Ad Block Plus blocks the communication. In consecuense, the source that Ad Block Plus has blocked can not load content in the web site or tracks the user.

However, if a user feels safe in a concrete site or does not care about to be tracked, user has the possibility to allow this web site, so Ad Block Plus does not interfere in the communication.

In 2011 they introduced a white list which contains the domains that Ad Block Plus allows. They created this list because they realized that the ads from these site are not intrusive or because they are not a threat to user's private data. Nevertheless, there is an important controversy around the

[1]Ads are considered as intrusive ads if they hide contect or it they pop-up randomly on the screem making user experience annoying

whte list. Ad Block Plus started to accept money from some ad companies, like Google, Microsoft or Amazon to move their domains from the list of damains to block to the white list [6].

3.3.2 Privacy Badger

Privacy Badger[2] is a browser plugin developed by the EFF.

Privacy Badges is based on Ad Block Plus. It blocks the traffic going to the advertiser and to any other third-party trackers used by websites which may cause a privacy harm to the user. Privacy Badger use an algorithm to detect the trackers. As I have explained, website is made up by different sources. When a user visits different web sites, Privacy Badger detects if all these sources, that are part of the site, follow the user across different sites. In the case that Privacy Badger discovers that someone is tracking this user, the tool blocks the communication with this source so If the communication is blocked this source can not track the user.

So, this functions are completly automatics and theres no possibility to make differences between trackers. Privacy Badger considers all the trackes as elements to block, however, in the Ad Block Plus, there is a preferencial treatment with all these companies that are paying to appear in the white list. So, the users feel that nowadays the Ad Block Plus goal is make money and not protect the users.

Privacy Badger has a high number of false positives and false negatives. It happens because Privace Badger's goal is protect the user's from trackers, not block ads. This tool allows ads but only from the sources that do not follow the users across different website. So, if the source that provides ads is not interested in track users, Privace Badger will not block the communication, so the ads will appear in the site like in a normal session.

This is a motivation for the advertisers to change the policy that they are following in this moments. Ad companies not only offer ads in websites, they track users to extract private data from them to create customized ads

[2] `https://www.eff.org/es/node/73969`

according to their behavioural trait [2.2]. If all user use tools like this one, the only way that advertisers can offer their products is crating good ads to the users. Which means offer products and services but not tracking users.

3.3.3 Ghostery

Ghostery [16] is probably the most important anti-tracker in the market nowadays.

3.3.4 NoScript

NoScript [17] is a free browser pluging for Firefox developed by Giorgio Maone [18]. It was considered as one of the top 100 products of 2006 according to PC World's [19]. NoScript blocks all the JavaScripts, Java applets and Flash elements embedded in web sites. As it can also block other plugins, users started to move towards less aggresive solutions like adblockers.

3.3.5 MyTrackingChoices

MyTrackingChoices is a browser plugin for Google Chrome. This plugin has been developed for a research study of INRI Grenoble [20] that is a public science and technology institution dedicated to computational sciences.

It is classified as anti-tracker and the aim of this plugin is give the opportunity to users to select in which kind of web sites they do not want to be tracked. To get this goal they have developed a categorization algorithm based in the methodology [21] and the Firefox Interest Deshboard plug-in. With this algorithm the system knows the topic of the web site. Furthermore, the plugin offers a list of categories where users can select the categories that they do not want to be tracker. When the user visits a web site, the system extract the topic using the categorization algorithm and if the topic is equal to the categorization that the user wants to block, the plugin starts to block trackers.

Furthermore, the plugin offers the users the possibility to allows web sites, independently of the topic that they want to block.

In summary, the goal of this plugin it gives users the posibility to choose the topics where they prefer to surf safely.

3.3.6 Project Contribution

As opposed to Ghostery [3.3.3], Privacy Badger [3.3.2] and MyTracking-Choices [3.3.5], I am not interested in blocking ads but in protecting user's privacy. The main difference between my plugin and the state of the art is that my plugin gives users the opportunity to choose the trackers that they want to block and in which domains want to enable tracking activity. If the user does not block any tracking activity on a given website or to a given tracker, the plugin will block any network comunication. I also extend my development efforts with a research study that analyzes how online advertising uses web cookies to build user profiles that define the advertisements that will be rendered to the user.

Chapter 4

Plugin Implementation

In this chapter, I describe the different use cases in which both a user and a developer may interact with the plugin, the design of the plugin and its implementation details.

4.1 Use Cases

In this section I describe the typical use-case for the plugin. In that scenario, an actor is an identity that interacts with the system without being part of it. There are two principal actors in this architecture

- **User:** Represents the user who interacts with the plugin.
- **Developer:** Represents an external human who has access to the system.

Below, I provide the template that I have used to define and characterize each one of the 7 use cases.

Then, I will explain the different use of cases that appear in Figure 4.1.

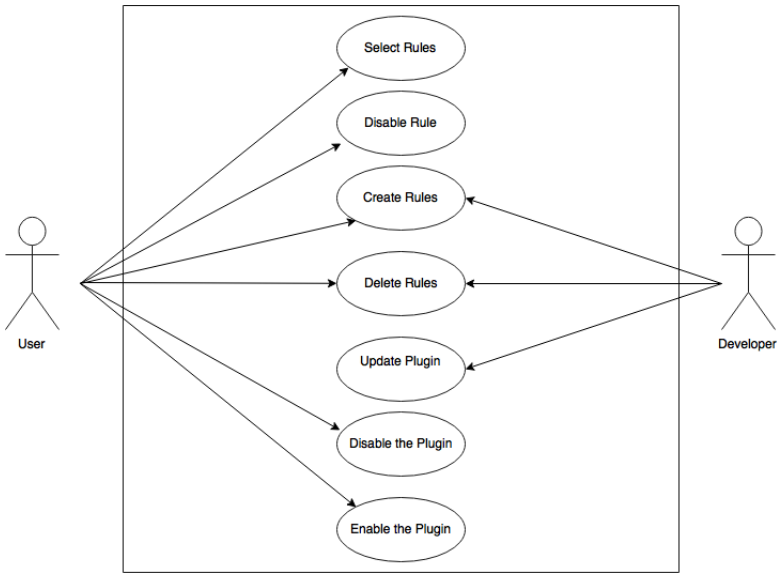

Figure 4.1: Use cases

Identification	Represents the Id of the Use of Case. I indentify each use case with the nomenclature US-XX, where XX represents a positive number
Use of Case	Name of the use of case
Primary Actor	Actor that appears in the process
Brief	Description of the use case
Preconditions	Conditions that have to appear to start the process
Postconditions	Effects in the system when the process completes
Exceptions	Events that appear in the case that something is wrong during the process

Table 4.1: Definition of a use case.

Identification	UC-01
Use of Case	Select Rule
Primary Actor	User
Brief	User selects the rule in the web site. The plugin starts to block the traffic from the domain that the rule establishes
Preconditions	The user has to be already logged in the system. The connection between the plugin and the web site has to work correctly
Postconditions	The plugin starts to apply the rules that the user has selected
Exceptions	The conexion between the plugin and the web site does not work correctly. The plugin does not use a new rule that user has selected

Table 4.2: UC-01

4.2 Software Design

4.2.1 Sofware Architecture

The system architecture follows a "Clientserver" model. In this model there are three different modules:

Identification	UC-02
Use of Case	Disable rule
Primary Actor	User
Brief	The user disables the rule in the web site
Preconditions	The user has to be logged in the system. The connection between the plugin and the web site has to work correctly
Postconditions	The plugin disables a rule. The plugin does not apply the rule
Exceptions	The connection between the plugin and the web site doesn't work correctly. The plugin is still uses an old rule

Table 4.3: UC-02

Identification	UC-03
Use of Case	Create rule
Primary Actor	User and Developer
Brief	The user or developer creates a new rule in the web site
Preconditions	The user has to be be logged in the system Furthermore, the rules should not already exist
Postconditions	The plugin starts to apply the rules that the user has created
Exceptions	The connection between the plugin and the web site does not work correctly. The plugin does not apply the new rule that user has created

Table 4.4: UC-03

Identification	UC-04
Use of Case	Delete rule
Primary Actor	User and Developer
Brief	The user or developer wants to delete an existing rule
Preconditions	The rule has to appear in the database
Postconditions	The plugin deletes the rule and starts allowing traffic previously blocked by the deleted rule
Exceptions	The button does not work. The plugin is not capturing the traffic

Table 4.5: UC-04

Identification	UC-05
Use of Case	Update plugin
Primary Actor	Developer
Brief	The developer inserts new features or updates the system according to the new browser version
Preconditions	–
Postconditions	The developer changes the plugin
Exceptions	The upgrade is not compatible with the browser version run by the user

Table 4.6: UC-05

- Server: The server-side provides the rules that user wants to apply in its plugin.

Identification	UC-06
Use of Case	User disables the plugin
Primary Actor	User
Brief	The user wants to disable the plugin in the browser
Preconditions	The plugin has to be installed and enabled in the browser
Postconditions	The plugin does not capture and block the traffic that the user generates
Exceptions	The button does not work. The plugin is still capturing the traffic even after being disabled

Table 4.7: UC-06

Identification	UC-07
Use of Case	User enables the plugin
Primary Actor	User
Brief	The user wants to enable the plugin in the browser
Preconditions	The plugin has to be alredy disabled in the browser
Postconditions	The plugin starts to capture and block the traffic that the user generates according to the existing rules.
Exceptions	The button does not work. The plugin is not capturing the traffic

Table 4.8: UC-07

- API: This module, which I implemented on the client side, enables interaction with the server API that contains the rules. The server rules are stored on Wedia's servers – one of the project partners. The goal of this module is establishing the communication between the server and the client. The API reads the rules and sends them to the blocker module.
- The Blocker: This module is part of the client side. When API module provides to the blocker module the rules that user have selected, the blocker module starts to block the traffic according to the user's rules.

4.3 Detailed Design

4.3.1 Server Module

The server is the provider of the resources. In my case, the server provides the different rules that the user wants to apply. The data is stored in Wedia [9] servers according to the european law of data protection [3.1.1]. This module

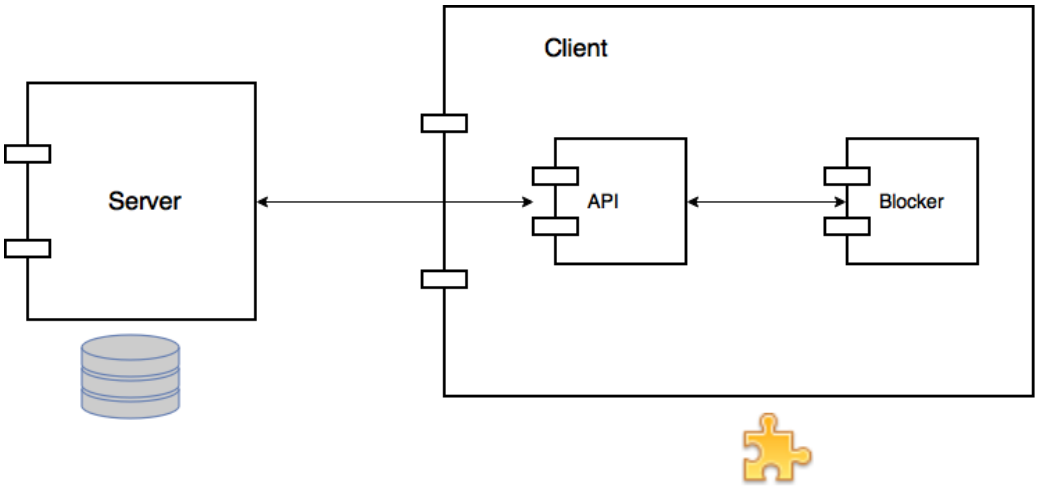

Figure 4.2: Plugin Architecture

can communicate with the client using the protocol Oauth2. I provide an explanation of the communication process between the client and the server in the next subsection.

4.3.2 API Module

Users have the possibility to select and create their own rules. To achieve this goal, users can use a web-based graphic interface. In this site the users can select the rules that they want to apply in their plugin and create new rules if they consider that it is necessary for them to protect their own private data. All these rules are stored in Wedia's servers, so I need to establish a communication channel between the plugin and the servers storing user rules. This communication is built on top of the *Oauth2* authorization protocol.

The protocol follows some steps that I have to establish in my implementation as shown in Figure 4.3.

1. **Authorization Request:** When users download the plugin, the first thing that they have to do is registering themselves in the website.

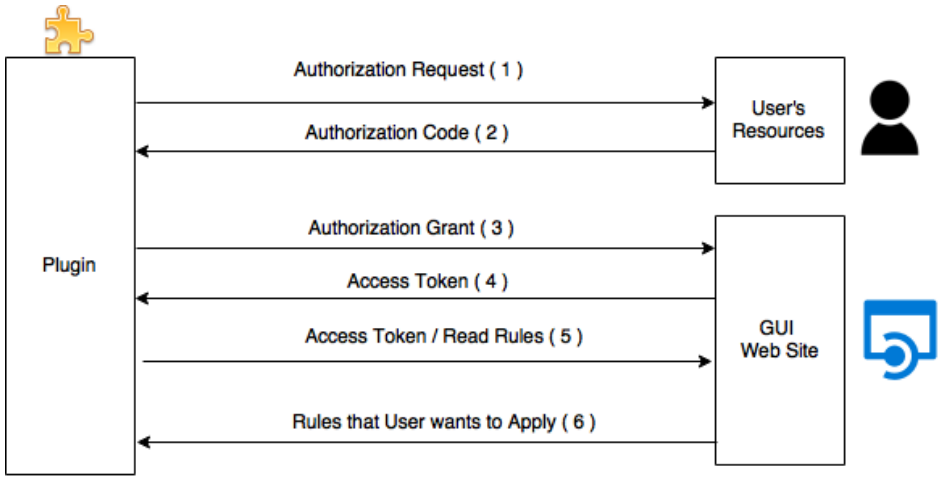

Figure 4.3: Oauth2 Protocol

Then, they have to approve the access to their resources, i.e., their blocking rules.

2. **Authorization Code:** When users authorize access to their rules, the plugin reads the *authorization code* that it receives from the server.

3. **Authorization Grant:** Once the authorization code is downloaded, the plugin needs to obtain its accesToken which allows the plugin to establish the communication.

POST /resource HTTP/1.1
Host: typesbackend.newlab2.wedia.gr/oauth/access_token

{grant_type:authorization_code, client_id:client_id,
client_secret:client_secret, code: *codeAccess*, redirect_uri:getcode}

4. **Access Token:** If the previous steps are successful, the client receives an *Access Token* and a *refresh_token:*

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{access_token:_*token*,expires_in:time,refresh_token:token}

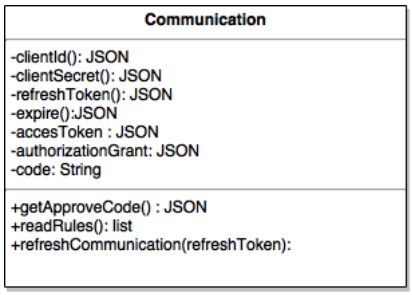

Figure 4.4: Design API Module

5. **Access Token / ReadRulesAPI:** When plugin receives the approval to read user's rules, it sends the acces token in the header of the request.

GET /resource HTTP/1.1
Host: typesbackend.newlab2.wedia.gr/api/list
Authorization: Bearer accesToken

6. **Read Rules:** The response from the web site contains the rules that the user wants to apply in its plugin. The plugin reads the rules and starts to block traffic according to them.

In this module there is only one class *Communication* that provides 4 methods:

- **getApproveCode:** This method detects when a user visits the web that enables configuring user's rules.
- **readRules:** When users grant the plugin access to their rules, this method establishes the communication channel between the plugin and the server using the Oauth2 protocol as shown in Figure 4.3. Once this communication is successfully established, the plugin recieves the user's rules from Wedia's server.
- **refreshCommunication:** The Oauth2 protocol implements a session timeout for security reasons. It is therefore necessary to refresh the channel to avoid this situation from happening by sending a "refresh token" to get a new acces token when this sittuation occurs. It

is important to say that it is not possible to establish a real time communication channel with the server, so when the API refreshes the channel, the API needs to synchronize and merge any changes in the rules between the plugin and the server.

Below, I list the predefined rules that my plugin offers to its users by default:

Site	Tracker	Action
*	doubleClick.net	block
*	cxense.com	block
*	ad-srv.net	block
*	third-party CookiesFacebook	block
*	third-party CookiesTwitter	block

Table 4.9: Predefined rules

In this case, the plugin blocks all communication going to those trackers from any website. As aforementioned, the user can customize and extend these rules.

4.3.3 Blocker Module

The client in this model is the plugin that I have developed. The first one, "Oauth" comunicates with the server and fetches the rules that user wants to apply. The "Blocker", which is part of the plugin, applies the rules as it intercepts and analyzes web traffic. It is composed by 5 classes which all together provide the required functionalities to block tracking traffic.

Index Class This is the principal object of the blocker module. It implements three methods: *initiate*, *button* and *addObserver*.

When the user downloads the plugin and installs it in the browser, the method *initiate* redirects the user to the web site [**?**] to create a new profile – in case the user does not have previously registered, otherwise it logs the user in the service – and establishes the comunication between the plugin

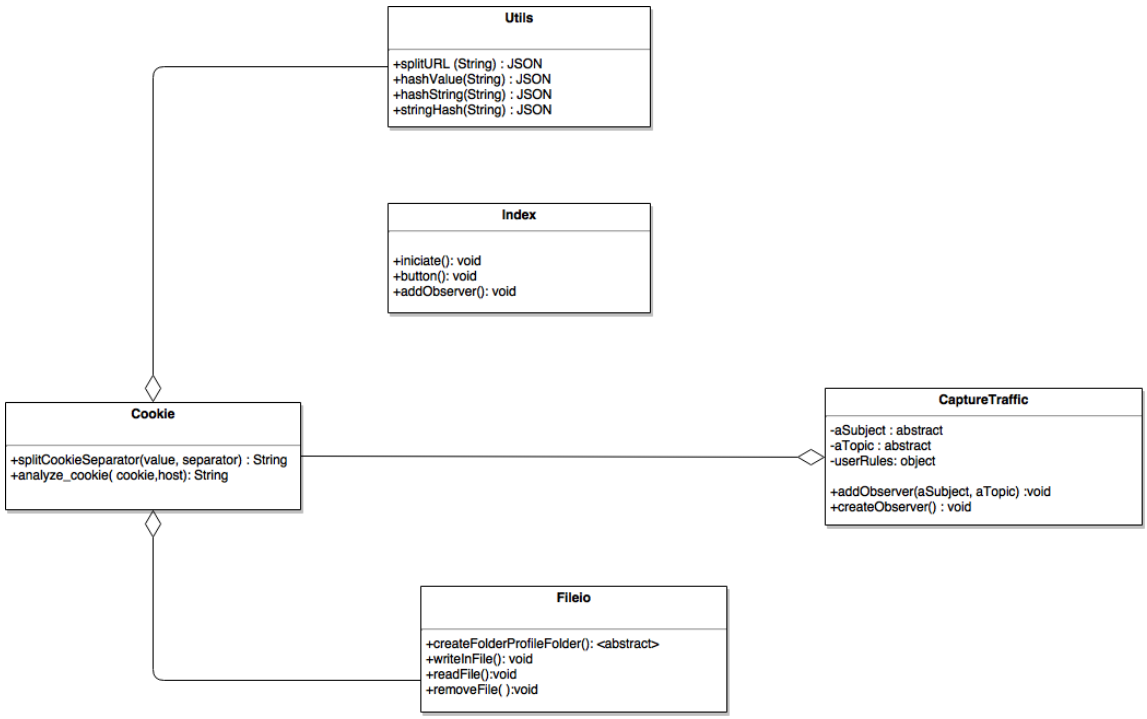

Figure 4.5: Blocker's design

and the web site hosting the rules. Whenever the user clicks in the plugin's button which appears in the browser's toolbar, which is controlled by the the method *button*, the plugin opens a new browser tab and redirects the user to the control panel where he/she can enable, create or disable blocking rules. Finally, this class also creates a new instance in the *captureTraffic* class using the method *addObserver*. The observer will capture and modify the traffic generated by the user according to the rules enabled at a given time. I create two observers, one that captures the HTTP requests before forwarding them to the Internet and another responsible for capturing the responses.

CaptureTraffic Class This class implements a single method that captures and modifies the traffic according to the rules. To achieve this goal, it is neccesary to create two observers which connect to the Firefox API that enables capturing browser's traffic. Depending on the argument passed when invoking this method, the observer will capture the response or the request:

- *"http-on-modify-request":* In this case, it is called whenever an HTTP request is made. The network channel is available and allows me to modify traffic content such as HTTP headers and payload.
- *"http-on-examine-response":* This method is called when an HTTP response has been received from the web server.

Utils Class This class provides functionalities parse and manage web cookies and the traffic captured to analyze the traffic captured by the plugin. This class implements 4 methods. However, the principal method is *splitUrl* which parses the information embeded in the URL (passed as an argument) visited by the user.

This method returns the information in a JSON format like this:

$$\{protocol{:}protocol,host{:}host,parameters{:}parameters\}$$

The remaining methods perform casting features and data format operations – e.g., to analyze hashes or Strings.

Cookie and Fileio Classes These two classes expand the plugin functionalities which I require for the future work that I describe at the end of the dissertation, in particular for analyzing and transforming web cookies.

Chapter 5

Analysis of the impact of web cookies on OBA

In this section, I present the tools that I have developed to study the effects of HTTP cookies in OBA and the methodology that I have followed to achieve the second goal of my project [1.2].

5.1 Selecting the Set of Training and Control Pages

I use the tool AdWords, a service from Google Inc. that provides demographic information about their potential customers (e.g., age, gender or topics of interest) and the information about web sites to advertisers. In particular, AdWords classify websites in 2000 different categories. For each category, AdWords provides a set of representative websites.

I have developed the "Persona" concept. The persona is a training set of websites based on the top ten most representative web site for each category. The goal is that a "persona" will have a clearly defined interest in a topic and therefore, it will present a specific behavioural trait. Obtaining very clearly defined personas is fundamental in this study as if a given persona contains noise – i.e., interest in more than one topic – it will be almost impossible to

Figure 5.1: Venn Diagram Personas

identify its behavioural trait.

Each persona in the Venn's diagram illustrated in Figure 5.1 has a set of topics represented by the small circles. In the first group, we can see that the different personas share the same topics, so a tracker would find similarities between them and the ads delivered to this set should be similar. The users falling in a different personas should obtain a different behavioral trait and the ads that appear in their relevant websites should be completely different.

I also define a set of control websites, completely neutral to the personas, for the experiments. Each set contains 7 websites [1] about weather so they should be completely independent of the topics of interest for each persona. The idea is that, after visiting a website like *https://www.marca.com* – an important Spanish sports newspaper – with the "sports persona", the ads rendered in *www.wunderground.com* – a weather website – should be related to the behavioural trait of the user, in that case, about sports. Appendix I lists the 120 personas defined for this project.

[1] www.wunderground.com, www.eltiempo.es, www.tutiempo.net, www.tiempo.com, www.weatherbase.com, www.accuweather.com

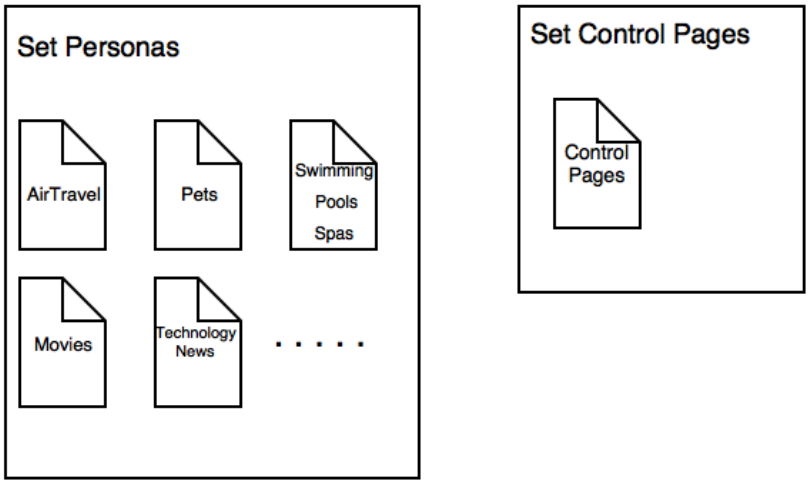

Figure 5.2: Set Personas and Control Pages

5.2 Crawler Experiment

A web crawler is a bot – i.e., a piece of software that simulates a human user surfing in the web – that fetches websites automatically. I developed a custom web crawler using Python and Selenium WebDriver framework [2]. Selenium provides developers the ability to drive or interact with a browser as a user would do to achieve more relism (as websites can block requests if they detect that they are originated from a bot) and to fetch the web site source code to extract information embedded in the HTML.

I use the crawler to fetch the websites required to create the behavioural trait for each one of the personas defined in the previous section [5.1] and to extract information from the control pages source code, in my case, the ads embedded in the HTLM.

The crawler implements 5 methods:

- **main:** This method starts the program and loads the set of personas *listPersonas* from a number of text-files: e.g., *movies.txt*, *airTravel.txt* and so on. Then, for each persona in the *listControlPages* I call the

[2] `http://www.seleniumhq.org/projects/webdriver/`

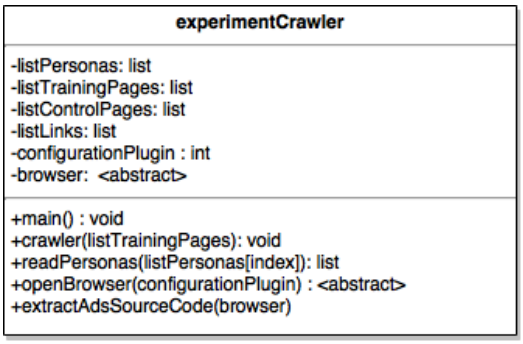

Figure 5.3: crawlerExperiment Design.

crawler method to fetch the websites and run the experiment for each persona.

- **crawler:** This method calls the *openBrowser* to create a new Firefox session. Also, I read the list of training pages (i.e., websites present in the persona that I have selected in the main method), using the method *readPersonas*. Then, when I create the new Firefox session and I extract the web sites from a concrete persona, I visit each web site to create its corresponding behavioural trait. To create the web fingerprint for each persona, I visit the training pages for each persona during 900 iterations. For each 10 iterations I extract the ads from the control pages (i.e., the neutral ones) using the method *extractAdsSourceCode* in order to see the ads printed in them. The image 5.4 shows the flowchart of each experiment.

- **openBrowser:** This method creates a new browser session which also contains a modified version of my plugin with an specific configuration in order to achieve the second goal [1.2] of my project. The only difference between this experimental version and the final one is that the former does not need to establish any communication with the web site and the plugin to obtain the set of rules: they are pre-configured directly on the source-code as there are no user-defined events in that scenario. The rules are defined in Table 5.1. This allows me to see

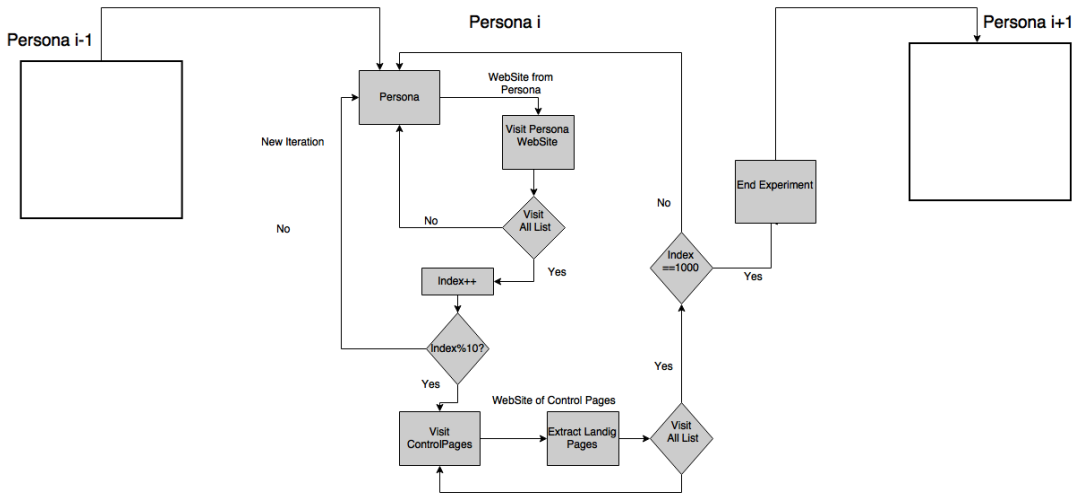

Figure 5.4: Flowchart Extract Ads From Control Pages.

how HTTP cookies affect in OBA – i.e., all the configurations that I establish which can block or allow the HTTP cookies according with the hardcoded rules. I select the configuration of the plugin when I create the new browsers session. Finally, I return the browser element to use it in the *crawler* method with the plugin installed in it.

Id	Configuration	Action
0	allCookies	allow
1	allCookies	block
2	thirdCookies	block

Table 5.1: Predefined rules experiment version.

- **readPersonas:** This method loads and parses the name of the persona and the relevant websites, which will define the persona's behavioral trait, from its corresponding file (e.g., *movies.txt*). The image 5.5 illustrates the information that this method reads and its output format.

- **extractAdsSourceCode:** This method extracts the ads from the source-code fetched from the control pages.

  Figure 5.6 represents an example of a control page's source code. All the ads are embedded in *iframes*. Tipically, an *iframe* is used to embed

```
http://space.com
http://naked-science.ru
http://sibirica.su
http://linkstars.ru
http://cosmos-online.ru
http://artemjew.ru
http://space-x.ru
http://rosregistr.ru
http://mapgroup.com.ua
http://selena-luna.ru
```

Figure 5.5: Example of persona.

another document within the current HTML document, in this case, it is used to embed ads [25]. However, the part of the source code that I need to extract is in the *a* tag which defines the hyperlink used to point in order to extract the landing page of the displayed advertisement.

To gather these hiperlinks I need to visit all the iframes that I find in the source code. Sometimes, the information that I want to obtain is embeded in nested iframes. For this reason, I establish an Iterative deepening depth-first search with a maximun depth of 4.

The image 5.7 represent the algorithm that I have developed to find the ads. During my observations, I realized that ads usually appear in iframes with depth 2 or 3. Therefore, the threshold of 4 levels is sufficient. Finally, I return the list of all the ads that I have found in this iteration.

The first experiment finishes when I visit during 900 iterations all the persona's web sites and I extract all the ads that I find from control pages. Then, I close the firefox session to delete the persona, I select another persona and I start again the same proccess with another firefox session.

```
1  <!DOCTYPE html>
2  <html>
3    <head>..</head>
4    <body>
5      <iframe id="google_ads_frame2"src="https:/
          googleads.g.doubleclick.neth=280&am&amp;dtd
          =1336" marginwidth="0"></iframe>
6      <html>
7        <head>...</head>
8        <body>
9        <a id="aw0" target="_top" https://www.
           googleadservices.com/pagead/
           nJcIhV7yJSq9QjAndv97500\& adurl=http://www.
           britainsfinest.co.uk/hotels>
10       <img src="https://tpc.googlesyndication.com/
           simgad/8810562865345994243" border="0" width
           ="336" alt="" class="img_ad" onload=""></a>
11       </body>
12       </body>
13   </html>
14 </iframe>
15 </html>
```

Figure 5.6: Example of Control Page HTML

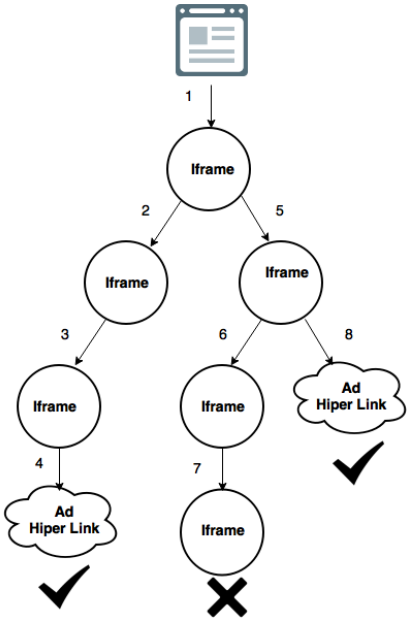

Figure 5.7: Iterative deepening depth-first search

5.3 Data Sanitization

It is important to sanitize the data that I have obtained from the control pages in the step before [5.2]. This data contains a lot of noise that I need to remove and extract parts from it that are essential in the study.

This is an example of the data that I have obtained in the step before. However, the interesting part of the data is the landing page which represents the ad embedded in a control page.

My aim it is to transform the hiperlink:

https://www.googleadservices.com/pagead/nJcIhV7yJSq9QjAndv97500&
*adurl=http://***www.britainsfinest.co.uk***/hotels*

In the following URL which is a more meaningful representation of the ad's landing page:

www.britainsfinest.co.uk

```
from selenium import webdriver
def main():
  var i=0
    while(i<len()):
           crawler(listPersona[i],plugingConfiguration)
           i++
def crawler(listPersona[i],plugingConfiguration):
    listTrainingPages=readPersonas(listPersona[i])
    listControlPages=readControlPages()
    browser=openBrowser(plugingConfiguration):

    while(iteration<900):
        for(webSite in listTrainingPages )
                browser.get(webSite)
            if(iteration%10):
                for(webSite in listControlPages)
                    browser.get(webSite)
                    extractAdsSourceCode(browser)
                    //Surfing in the source code//
                fileOut.write(listLinks)
            else:
            iteration++
        browser.close()
if __name__ == "__main__":
    main()
```

Figure 5.8: crawlerExperiment Code

```
adurl=((http)|(https))[(\w)|(\%)]*(://)*(www)*[\.]*[(\w)|(\-)|(\.)]
url[(%)|(\w)]*((http)|(https))://[www\.]*(\w+)\.(\w+)[((\w)|(\/)|(\
```

Figure 5.9: Regular Expresions Extract Landing Pages I

```
https://[(\w)|(\.)]*/[(\w)]*
(http|https)://www\.(\w+)\.(\w+)[((\w)|(\/)|(\.)|(\-)]*
```

Figure 5.10: Regular Expresions Extract Landing Pages II

To do this, I developed a python program that cleans the data using regular expresions. I have created the regular expresion according to my observations.

In the hyperlinks, the landing pages are always embedded in two different parts:

So, if the hiperlink that I have extracted from the [5.2] contains information that matches with one of these two regular expresions, is selected as candidate to contain a landing page. Furthermore, I delete the rest of the information that does not match with the regular expresion.

Then, the candidate to contains a landig pages has to pass the next test. If these candidates have matches with these regular expresions it means that they have a landing pages embedded in the data. Finally, I extract the landing page completed clean using this regular expresion:

Finally, I store all the landing page separate by the iteration where ad has been captured, like in the figure 5.12.

```
www\.(\w+)\.(\w+)[((\w)|(\.)|(\-)]*
```

Figure 5.11: Regular Expresions Extract Landing Pages III

```
Index0
www.stellenanzeigen.de
www.sharethis.com
www.reachgroup.com
Index1
...
...
...
Index90
www.vloors.de
www.ad4mat.de
www.ems.ch
www.steiermark.com
```

Figure 5.12: Example of landing pages extracted from Control Pages.

5.4 Extracting Topics from Ads Using an adWords Crawler

Now, I obtain information about the topics of the ads that I extracted with the crawler [5.2] from the landing pages in the last 5 interactions [Figure 5.12] so that the persona is estable and clearly defined. Once more, I implemented the crawler in python and Selenium WebDriver.

The first step in this part of the methodology, is extracting the landing pages and store them in a list. Then, the crawler visits AdWords using Firefox browser, it logs in with an account that I have created for the experiments and it surfs up to the part where I find web site that I am interested[3], in this case, the part that provides information about the topics of a web site. When the crawler is in this part of the website, I extract the principal topics of the landing page and the relevance of them in the site. The crawler obtains the data from the source code of the AdWords and store it. Finally, the crawler refreshes the site and extracts information from another landing page that

[3] `https://adwords.google.com/da/DisplayPlanner/Home?__u=5318450477&__c=1225666757&authuser=2#results`

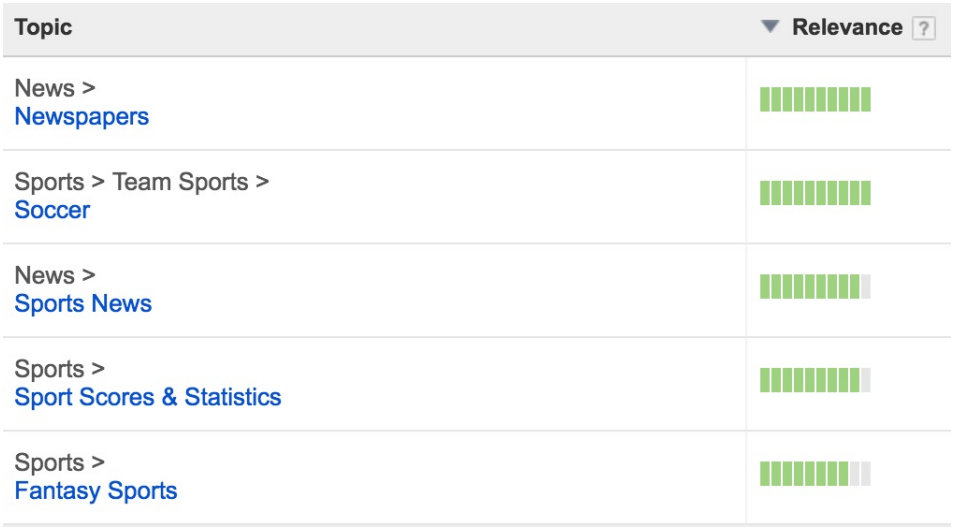

Figure 5.13: Example AdWords Information.

appears in the list until there are no new landing pages to visit.

The image 5.13 is an example of information that I extract from AdWords. This is the result for the domain **www.marca.com**:

Figure 5.14 shows an example of how I store the information that I have obtained with the crawler. For the landing page *www.quantcast.com* there are different topics which are part of the web site and the relevance of the topic. Furthermore, I count the number of times that this landing page appears in the set of landing pages.

5.5 Data Analysis

In this step of the methodology I measure the presence and the level of OBA in the experiments that I have performed. To measure the level of OBA it neccesary use meaningful metrics:

```
www.quantcast.com 3
Internet & Telecom > Web Services >
Web Stats & Analytics 10
People & Society > Social Sciences >
Demographics 10
Internet & Telecom > Web Services >
Search Engine Optimization & Marketing 9
Internet & Telecom > Web Services >
Affiliate Programs 9
Internet & Telecom >
Web Apps & Online Tools 8
Internet & Telecom > Web Services >
Web Design & Development 8
Business & Industrial >
Advertising & Marketing 8
Internet & Telecom > Web Services >
Web Hosting & Domain Registration 8
Business & Industrial > Advertising & Marketing >
Marketing 7
Internet & Telecom >
Search Engines 7
Internet & Telecom > Service Providers >
ISPs 7
Internet & Telecom >
Web Services 6
```

Figure 5.14: Topics of ads from AdWords

- I define the set of unique keywords associated with the training pages for a persona on source.

$$K_{T_{ps}} \tag{5.1}$$

- I define the set of unique keywords associated with the landing pages of ads shown to a persona p on source s on control pages.

$$K_{L_{ps}} \tag{5.2}$$

- I define the set of a that appear in a couple of iteratios of the experiments. The value of the experiment is the different configuration that the plugin could establish.

$$S_{c_i} \tag{5.3}$$

  where c is the different configuration that the pluging could establish and i the iteration that I extract the landing pages.

Targeted Training Keywords (TTK):

According to these metrics, I extract the presence of the topics relevant for each persona in each landing pages that I extracted from the control pages. The numerator is the number of topics in commun between the keywords associated with training pages and the keywords associated with control pages. The denominator is the set of keywords from the training pages.

$$TTK(p,s) = \frac{|K_{T_{ps}} \wedge K_{L_{ps}}|}{|K_{T_{ps}|}} \in [0,1] \tag{5.4}$$

TTK is a metric that appears in the study [23]

Variation of the Ads across the Time (VotAaT): This function represents the divesity of ads acrros the time. This value shows the size of the intersaction between the set of the landing pages in the first interactions of the experiment and the set of the landing pages in the last interactions of the experiment as the persona becomes more stable. So, if this value is higher, it

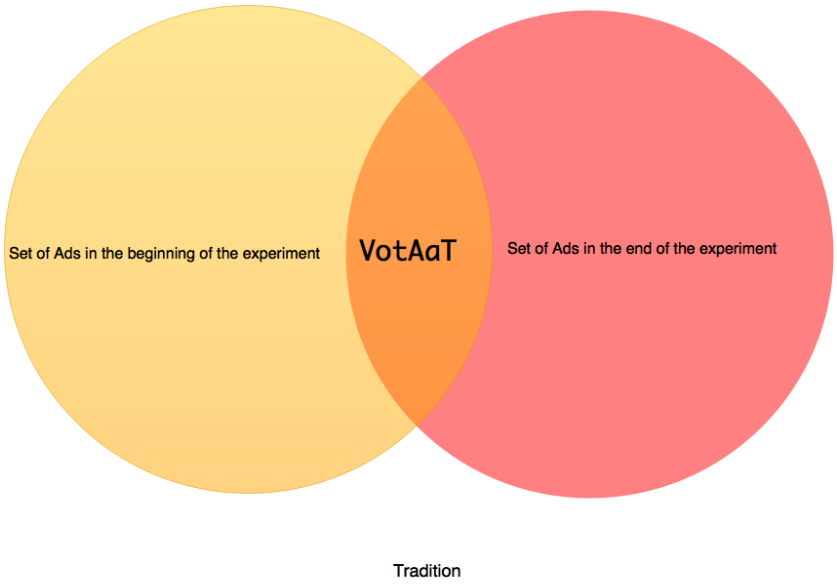

Figure 5.15: VotAaT.

means that the first interaction of the experiment has more or less the same landing pages that in the end of it.

$$VotAaT(c, f, l) = \frac{|S_{c_f} \wedge S_{c_l}}{|S_{c_f} \vee S_{c_l}|} \in [0, 1] \tag{5.5}$$

I show how big is the intersection between the ads from a concrete persona at the beginning of the experiment and at the end of it (and the presence of the same ad at as the persona evolves). If the value is close to 1, the ads did not change since the first iteration to the last one. However, if the value is close to 0 it means that ads changed during the experiment.

To measure the presence of OBA in the websites, I developed a python program that reads the data stored in the files – data about the topics of the ads landinga pages, like in the image 5.14 – and then I introduce them in the formulas that I have presented in this section. Finally, I plot the results that this formulas give me to analyze the results.

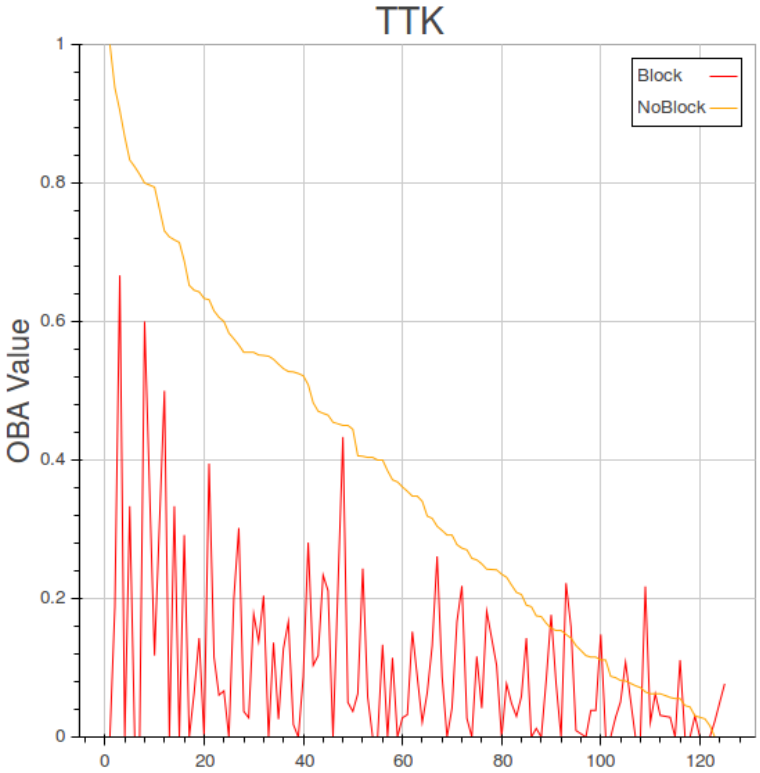

Figure 5.16: TTK value.

5.5.1 Results

TTK

In this section, I show the results of how the modification of the traffic affects – using the plugin that I have developed – in the ads printed in the web sites – in my case, the ads that appear in the control pages. In this study I only modify the HTTP Cookies and I take boolean decisions: either block cookies or allow them.

Figure 5.5.1 shows the value of OBA using the metric TTK. To generate this figure, I use all the topics form AdWords and I set topics, even if related as different as in the example below:

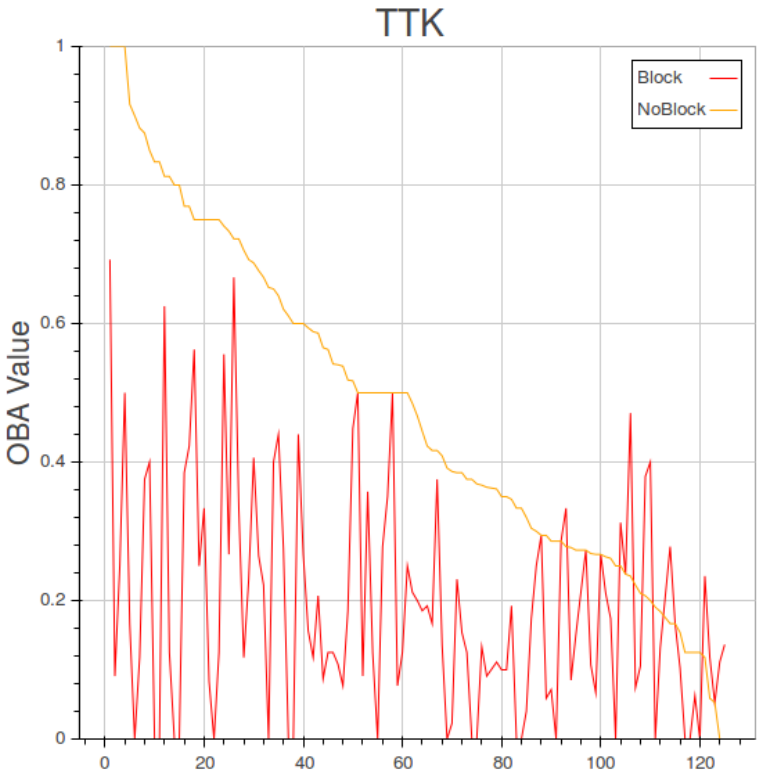

Figure 5.17: TTK value two levels.

Shopping-Consumer Resources-Loyalty Cards and Programs
Shopping-Consumer Resources-Computers

In Figure 5.5.1, instead, I use only the two main topics for each website as provided by AdWords. For instance, in the example below:

Shopping-Consumer Resources-Loyalty Cards and Programs

I only select the first two parts, in this case:

Shopping-Consumer Resources

As the Figures show, there is an important difference when I block the cookies: blocking cookies reduces completely the presence of OBA. This confirms that cookies are the driving force behind OBA.

To complement the results, I also created personas with a sensitive behaviour. As I defined in the background section, those are sensitive topics like politics, religion or sexual orientation:

AlzheimersDisease
CampaignsElections
DatingPersonals
DrugAlcoholTesting
Hinduism
InfectiousDiseases
SkepticsNonBelievers
DiscriminationIdentityRelations
MetalsMining
Cancer
LeftWingPolitics

According to the European laws, tracking should not happen in those personas. However, the empirical results demonstrate that tracking still occurs as shown in Figure 5.5.1.

According to the TTK value that I have obtained, OBA is present in 90% of the personas, even for personas associated with sensitive topics.

VotAaT

In this case, I have selected the 5 personas with a biggest difference between a profile that blocks cookies and another that allows them – according to TTK value.

The first five elements, from the 0 to 4, are profiles in which HTTP cookies have an important impact in OBA. Furthermore, the last 5 are profiles are HTTP cookies that do not have an impact in OBA.

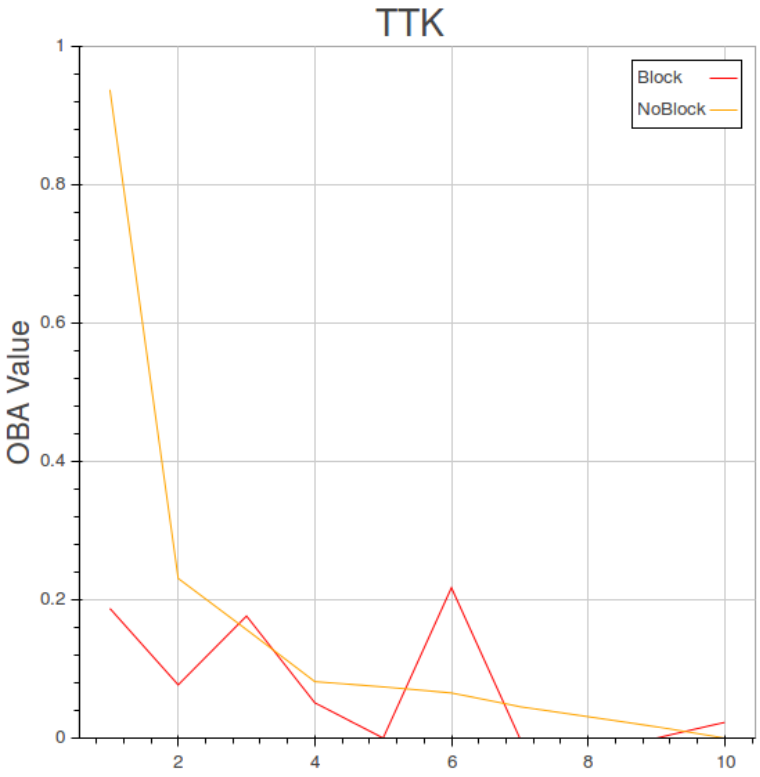

Figure 5.18: TTK value in sensitive topics.

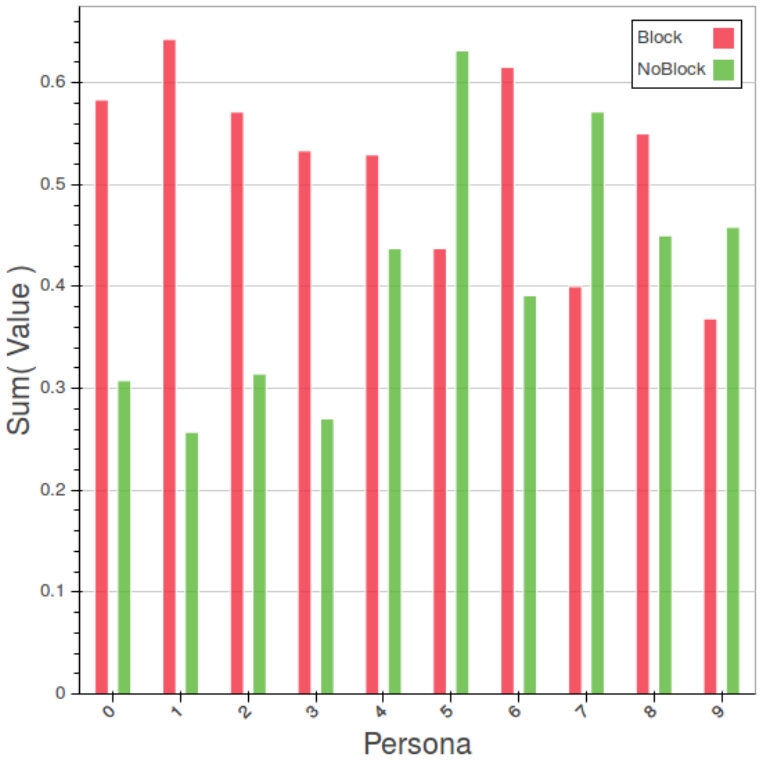

Figure 5.19: VotAaT results.

The results shown in Figure 5.5.1 show a strong correlation between the TTK value and the diversity of ads. The higher the TTK value is, the more diverse are the ads.

Chapter 6

Planification and Budget

In this chapter I will show the planification that I follow during this project and I will expose an economic analysis about this project.

6.1 Planification

The first thing in this project It's divide it in different sections which help me to organize better the project.

The project is divided in twelve sections:

1. **Study of the problem and definition of the project:** In this section I have extracted information about the related works in privacy and security in the networks and the different solutions that exist now in the market. Furthermore, when I have obtained the enought knowledge about the topic, I defined the project and the goals of it.
2. **Implementation of the plugin:** In this step I have developed the principle tool of this project. This tool is the browser plugin that captures and blocks the user's traffic –but always with user's consent–
3. **Plugin's testings:** When I finish to develop the plugin I need to test it to be sure that It works correctly. In this step I developed some test for the plugin.

4. **Implementation of Crawler:** I have developed the *experiment crawler* [5.2] using Selenium Webdriver for Python. With this crawler I am able to extract the data that I need for the study.

5. **Crawler's testings:** Like in the case of the plugin, It's neccesary to test if the *experiment crawler* works correctly.

6. **Desing Experiments:** In this step I have defined the experiments that I did to get the second goal of my project, study the impact of HTTP Cookies in OBA.

7. **Experiments:** I have run the experiments that I defined in the step before.

8. **Implementation of tools:** In this part of the project I did two tools. The first one, was *cleaningDatExperiments* [5.3] which is a program in Phython who cleans the data that I have obtained in the experiments using regular expresions[**??**]. Furthermore, I developed a crawler calls *adWordsCrawler*[5.2] that extracts information about the ads that I have obtained in the experiments.

9. **Tool's testing:** Like in the other cases, I test if everything works correctly.

10. **Analys of the results:** In this part of the project I developed a Python program [**??**] which uses the metrics [5.5] that I have defined to analyze the level of OBA.

11. **Documentation:** In this step I wrote the documentation of the project.

12. **Presentation:** Finally, I did the presentation of the project.

6.1.1 Initial Planification

When I have defined the different sections of the project, It's the moment to establish the initial planification of if.

Id	Activity	Start Date	End Date	Duration
1	Study of the problem and definition of it	01/12/2015	22/12/2015	3 Weeks
2	Implementation of the plugin	08/01/2016	05/02/2016	4 Weeks
3	Plugin's testings	08/02/2016	15/02/2016	1 Week
4	Implementation of Crawler	16/02/2016	01/03/2016	2 Weeks
5	Crawler's testings	02/03/2016	09/03/2016	1 Week
6	Desing Experiments	10/03/2016	17/03/2016	1 Week
7	Experiments	18/03/2016	15/04/2016	4 Weeks
8	Implementation of tools	21/03/2016	05/04/2016	2 Weeks
9	Tool's testing	06/04/2016	13/04/2016	1 Weeks
10	Analys of the results	18/04/2016	25/04/2016	1 Week
11	Documentation	26/04/2016	22/05/2016	4 Weeks
12	Presentation	25/05/2016	01/06/2016	1 Week
Total Weeks				**22**

Table 6.1: Initial Planification.

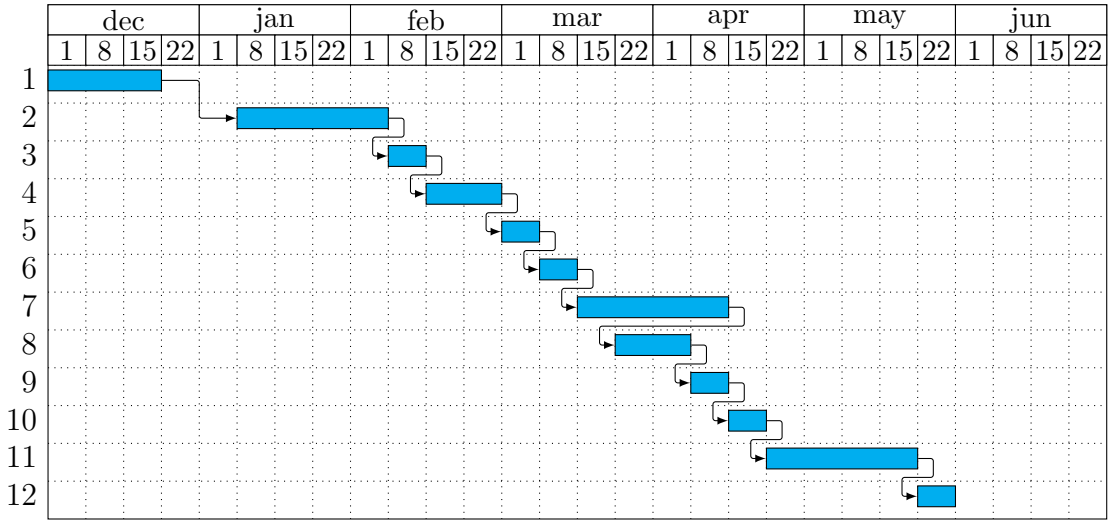

Figure 6.1: Gantt Initial Planification.

There were some sections of the project that were possible to did it at the same time. It was possible because in the step number 7 the experiments were running in different machines that I used to use for develop the programs. So, when the experiments were running I focused my work in develop the tools that I needed to study the effects of Cookies in OBA.

6.1.2 Real Planification

Id	Activity	Start Date	End Date	Duration
1	Study of the problem and definition of it	01/12/2015	22/12/2015	3 Weeks
2	Implementation of the plugin	08/01/2016	12/02/2016	5 Weeks
3	Plugin's testings	15/02/2016	22/02/2016	1 Week
4	Implementation of Crawler	23/02/2016	15/03/2016	3 Weeks
5	Crawler's testings	16/03/2016	23/03/2016	1 Weeks
6	Desing Experiments	24/03/2016	31/03/2016	1 Weeks
7	Experiments	01/04/2016	13/05/2016	6 Weeks
8	Implementation of tools	04/04/2016	25/04/2016	3 Weeks
9	Tool's testing	26/04/2016	03/05/2016	1 Weeks
10	Analys of the results	16/05/2016	23/05/2016	1 Weeks
11	Documentation	24/05/2016	14/06/2016	3 Weeks
12	Presentation	15/06/2016	22/06/2016	1 Weeks
Total Weeks				**26**

Table 6.2: Real Planification.

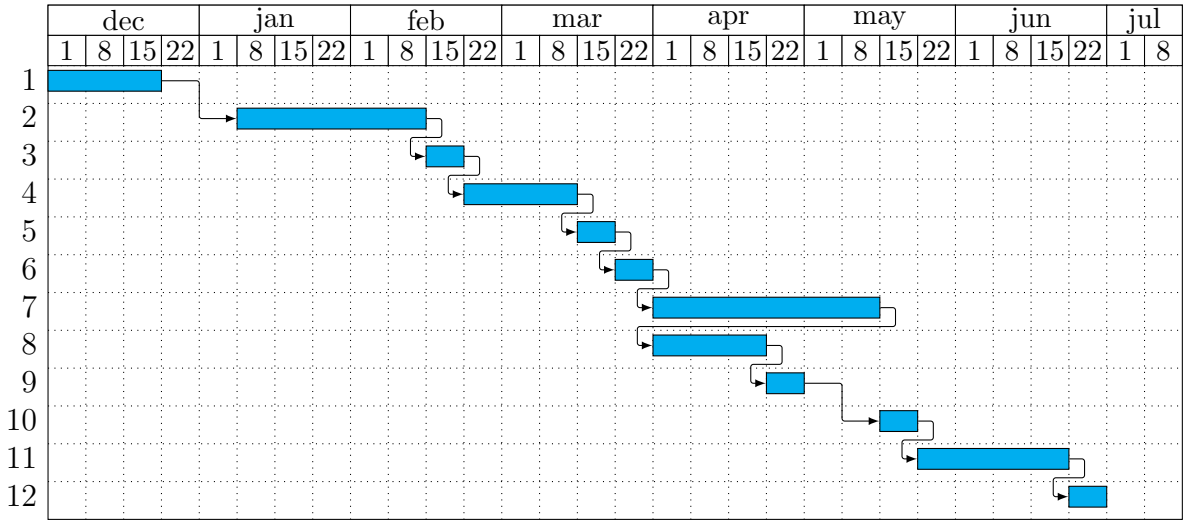

Figure 6.2: Gantt Real Planification.

If you compare the Real Planification with the Initial one, It's very easy to find an important difference. In the initial planification I established that I will work in the project for 22 Weeks. However, It wasn't possible. I have needed to expend more time in the experiments because each experiment took an important amount of time. Also, It was neccesary to repeat some experiments because I had some problems during them, like lost the internet conexion.

6.2 Economic Analysis

Before start the proyect, I designed an initial budget according to the initial planification. However, It was impossible to follow the initial planification. So, I have had a desviation in the final budget. In this section I will expose an economical analysis of the project, comparing the initial budget with the final one.

The first thing that I have to do is split the budget in differents sections:

- **Human resources:** Part of the budget for the workers involved in this project.
- **Hardware resources:** Part of the budget destined for the material that I need for this project.
- **Software resources:** Software necesary in this project.

6.2.1 Initial Budget

Human Resources

The human resources corresponds with the salary of the persons who were working in the project. In this case, I was the only worker. However, I have to include the ours that Dr. Ruben Cuevas Rumn expend supervising the project. In the initial planification I establish that It was necessary 26 weeks to finish the project. I have worked 8 hours per day and 5 days per week. It means that I have worked 40 each per week.

Worker	Salary	Total hours	Total
Pelayo Vallina-Rodriguez	5 €/h	880 hours	4.440€
Dr. Ruben Cuevas-Rumin	30€/h	100 hours	3.000€
Total			**7.440€**

Table 6.3: Human resources Initial Budget.

The salary that I establish for me, It's the salary that I recieved during my work in the project.

Hardware Resources

That's the Hardware neccesary to develop this project.

1. MacBook Pro
2. Hp Desktop Computer
3. Server Computer: CPU Intel Xeon E2680 2.5GHz with 2x12 Cores and 64GB of Memory.
4. Two NEC Desktop Display MultiSync E171M

Element	**Value**
1	1.449€
2	399€
3	999€
4	298€
Total Cost	**3195**

Table 6.4: Hardware Resources

Software Resources

1. Operating system
   (a) Ubuntu 14.04
   (b) OS X El Capitan
2. Programming language
   (a) Python
   (b) Javascript
   (c) HTML
   (d) LaTeX
3. Browsers

   (a) Firefox
   (b) Google Chrome
4. Integrated Development Environmen
   (a) Pycharm
   (b) WebStorm
5. Text editor
   (a) VIM
   (b) Sublime Text
6. Presentation
   (a) Keynote
7. Figures
   (a) draw.io

The total cost of the software that I used for this project was 0€.

Pycharm and WebStorm are IDEAs from JetBrains company. However, I was using a student license which gave me the opportunity to use them for free. Furthemore, Keynote is a program included in the Mac Pro for free.

Total Initial Budget

The initial budget It's the sum of the Human resources, Software resources and Hardware resources.

The initial budget is 10.635€. in total.

Concept	Total
Human Resources	7.440€
Software Resources	0€
Hardware Resources	3.195€.
Total Costs	**10.635€.**

Table 6.5: Initial Budget

6.2.2 Real Budget

Human Resources

There were some changes in the Real Planification that affected the final budget of the project. It was neccesary expend more time in some parts of the project. For this, the part of salarys in the budget was increased notably.

Additionaly, I required more supervision hours from Dr. Ruben Cuevas-Rumn. This has affected the final budget estimate.

Worker	Salary	Total hours	Total
Pelayo Vallina-Rodriguez	5€/h	1040 hours	5.200€
Ruben Cuevas-Rumin	30€/h	160 hours	4.800€
Total			**10.000€**

Table 6.6: Human resources Real Budget.

Hardware Resources

Hardware Resources was the same as I established in the initial budget, so in the end was neccesary to invest 3.195€in Hardware.

Software Resources

Like in the initial budget, all the software that I needed to develop this project was free. So the amount of money intended to the sofware is 0€

Total Real Budget

There is a difference between the Initial budget and the real one. The reason of this difference is becacuse It was neccesary to expend more time in the project that the initial planification. For these reason, I have to modify the Human resources acordding to the real planification.

Concept	**Total**
Human Resources	10.000€.
Software Resources	0€.
Hardware Resources	3.195€.
Total Costs	**13.195€**

Table 6.7: Real Budget

How can you observe, there is a difference between the Initial budget and the real one. The reason of this difference is that It was neccesary to expend more time in the project.

Desviation in the Project

It was necessary to invest **2.560€** more that the cost that I have established in the initial budget. All this money was invested in Human Resources,

Chapter 7

Conclusion and Future Sudies

In this project, I have presented a solution that seeks to find a balance between the online advertising industry and user's privacy in order to achieve a more sustainable web model. I have also presented a broad study of the impact of trackers on web ads, discovering possible violations of the European legislation when visiting websites of sensitive topics.

As future work, I would like to complete the development of a Chrome Version of the plugin to reach larger user audiences. The work is still in its early implementation phases. Nevertheless, I would like to conduct a deeper and more comprehensive study of the effect of OBA in today's website beyond the results presented in this study. We plan to submit our research findings to the 2017 World Wide Web Conference (WWW).

Bibliography

[1] How Unique Is Your Web Browser? *Peter Eckersley from Electronic Frontier Foundation (EFF)*

[2] https://downloads.pagefair.com/wp-content/uploads/2016/05/2015_report-the_cost_of_ad_blocking.pdf *The cost of AdBlocking*

[3] https://pagefair.com/blog/2015/ad-blocking-report/ *PageFair. The 2015 Ad Blocking Report*

[4] Narseo Vallina-Rodriguez,Rishab Nithyanand, Sheharbano Khattak, Mobin Javed, Marjan Falahrastegar, Julia E Powles, Emiliano De Cristofaro, Hamed Haddadi, Steven J Murdoch *Ad-Blocking and Counter Blocking: A Slice of the Arms Race*

[5] https://adblockplus.org/en/ *Blocker ad-Block Plus*

[6] http://www.engadget.com/2016/02/12/rip-adblock-plus/ *RIP: Adblock Plus*

[7] http://www.types-project.eu *Towards transparency and privacy in the online advertising business project*

[8] http://www.types-project.eu/index.php/partners/ *Towards transparency and privacy in the online advertising business's partners*

[9] http://www.wedia.gr/en *Wedia Limited*

[10] http://www.un.org/en/universal-declaration-human-rights/index.html *Universal Declaration of Human Rights (1948)*

[11] http://www.echr.coe.int/Documents/Handbook_data_protection_ENG.pdf *Handbook on European data protection*

[12] Paul Barford, Igor Canadi, Darja Krushevskaja, Qiang Ma, S. Muthukrishnan *Adscape: Harvesting and Analyzing Online Display Ads*

[13] https://acceptableads.org/es/ *Acceptable Ads. Mnifesto*

[14] http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32002L0058 *Directive on privacy and electronic communications of the European Parliament and of the Council of 12 July 2002*

[15] http://typesbackend.newlab2.wedia.gr/login *Plugin web site where user can select the rules that wants to apply*

[16] https://www.ghostery.com

[17] https://noscript.net

[18] https://addons.mozilla.org/es/firefox/user/giorgio-maone/

[19] http://www.pcworld.com/article/125706/article.html?page=7 *The 100 Best Products of 2006*

[20] http://www.inria.fr/en/centre/grenoble

[21] A. Kae, K. Kan, V.K. Narayanan and D. Yenkov. *Categorization of display ads using image and landing page features*

[22] Roberto Gonzalez, Hassan Metwalley,Miriam Marciel,Mohamed Ahmed, Ruben Cuevas and Saverio Niccolini. *Inside the Cookie Jar: Measuring the use of cookies in the wild.*

[23] Juan Miguel Carrascosa, Jakub Mikians, Ruben Cuevas, Vijay Erramilli and Nikolaos Laoutaris. *Always Feel Like Somebodys Watching Me. Measuring Online Behavioural Advertising.*

[24] Paul Barford, Igor Canadi, Darja Krushevskaja, Qiang Ma, S Muthukrishnan *Adscape: Harvesting and analyzing online display ads*

[25] http://www.w3schools.com/tags/tag_iframe.asp